Release Management Process

Mike Phillips
2011/11/03 14:14

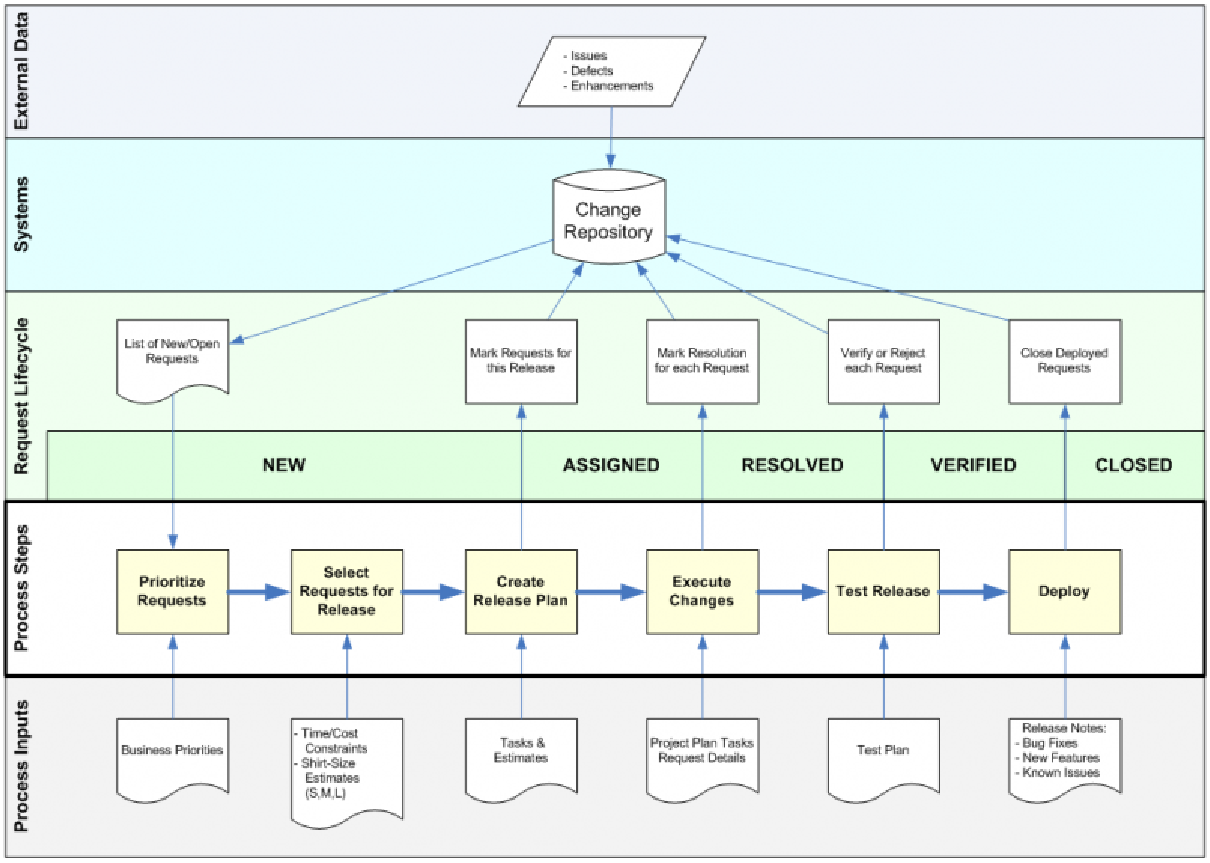
Table of Contents

Contents

Show

Process Diagram

Below is an illustration of the process (second-last row) and the inputs, outputs and effects of the process steps.



Process Steps

Prioritize Requests

The PM works with the customer to identify all of the requests (proposed changes) that are currently open (that is, not finished). There are two main sources for requests:

1. For new development, the **requirements** and resulting **project tasks**
2. Bug/Issue **trackers**

Each request should receive an individual priority rating from 1 (lowest) to 5 (highest priority), based on input from the customer and advice from the PM and/or technical staff.

Select Requests for Release

Technical staff should receive this prioritized list, and provide "shirt-size" estimates for each.

- **S (Small)** - 1 hour to 1 day of effort
- **M (Medium)** - 1 day to 1 week
- **L (Large)** - More than 1 week

Based on the rough estimates, the set of resources on the project, and any date constraints, the project manager and customer should create the list of requests that will be addressed in the release. Any associated tracker items should be marked with the appropriate target milestone (an arbitrary number defined by the PM or technical lead for tracking purposes). This will allow quick filtering for each release later in the process.

Technical staff should now take the list of requests for the release and create "10%" estimates - that is, estimates that are expected to be accurate within 10%. In almost all cases, estimates should be given in 1, 2 and 4-hour increments. If a given request is larger than 4 hours, it should be broken into steps of 4 hours or less and have each step estimated.

Create Release Plan

Using the list of requests and detailed estimates, the PM creates a project plan, with resource assignments, dependencies, and non-technical tasks (such as code migrations, security scans, code reviews, customer meetings, etc.). *Note: For frequent releases (> 2 per month), a project plan may be more effort than the work it tracks. In these cases, an Excel sheet with tasks, or a list of Trackers may be sufficient for coordination.* After the release plan is finalized, all associated tracker items should be marked as **ASSIGNED** and assigned to the staff responsible for delivery. At this point, the only unassigned (open) items should be those planned for a future release.

Execute Changes

The team should meet at least once prior to each release (or once per week, whichever is **more** frequent) to present status against the assigned tasks. The PM should keep track of the actual hours for each item, and compare this with the effort that was estimated. Project team members should share any issues that interfere with progress on their tasks, so that the PM is aware of them and can help resolve or escalate, if appropriate.

As each task is completed, the responsible team member **must** do the following:

1. Commit any changes (code, database, documents, etc) to version control, and reference the tracker number and/or project task in the commit comment.
2. Mark the tracker item as **RESOLVED**.
3. Update the tracker item and/or project task with the resolution of the item and (if appropriate) the number of hours spent.
4. Re-assign the tracker item to the PM unless otherwise directed.

Test Release

A Test Plan should be drafted for each release. The Test Plan may be based on previous versions but some effort should be made with each release to set the proper scope based on the changes that were made. The testing must include validation of all changes for the release. As the testing staff verifies each change against the system requirements and/or the tracker item, the tracker item should be marked as **VERIFIED** and re-assigned to the PM unless otherwise directed.

Note: Testing may find new issues, or determine that a given resolution was not sufficient or successful. In this case, the process should return to the Execute Changes phase, or the defective change should be rolled back so that the process may continue. The customer must be included in any discussion about changing the scope of a release.

Deploy

After the testing phase is complete, the PM and technical lead should prepare release notes with at least the following elements:

1. Release number (version number) and date, if applicable
2. Requests (issues, bugs, enhancements) resolved in this release
3. Known issues in this release
4. Configuration / environment changes required for deployment (config files, database changes, scripts to run, etc.)
5. Customer signoff

After the customer has signed off on the release, it can be deployed to production. The PM should coordinate activities with the project team and Operations staff to ensure that configuration and environment changes are applied during deployment.

When the release has been deployed to production, the PM should mark all applicable tracker items and project tasks as **CLOSED**.

Links

- Release Notes
- GForge Mailing Lists
- Project Management Life Cycle (PMLC)
- Software Development Life Cycle (SDLC)
- PMLC Related Documents